



В.П. Дерябкин

МОДЕЛЬ БАЗЫ ЗНАНИЙ ИНТЕЛЛЕКТУАЛЬНОЙ ФРЕЙМОВОЙ СРЕДЫ

(ФГБОУ ВПО «Самарский государственный аэрокосмический университет им. академика С.П. Королева (национальный исследовательский университет)»)

Качество и своевременность принятия решений в сложных ситуациях во многом определяется информационным окружением лица, принимающего решение. В [1 - 3] рассмотрена концепция построения интеллектуальной информационной компьютерной среды с фреймовой структурой знаний, которая позволяла бы создавать, развёртывать и исполнять интеллектуальные приложения в различных предметных областях. Ядром такой среды является база знаний с использованием фреймовых структур как унифицированных элементов представления знаний. Несмотря на то, что фреймовая модель представления знаний предложена давно [4], до сих пор формализация процесса решения задач вызывает значительные трудности [5,6].

В докладе рассмотрена и уточнена в соответствии с унифицированной моделью фрейма модель базы знаний инструментальной системы.

Фреймы базы знаний могут быть подразделены на группы:

- *активные* в логическом выводе результата, имеющие слоты с исполняемыми присоединенными процедурами (демонами и/или значениями слотов) F_a – поведение этих фреймов определяет логический вывод;
- *пассивные*, не имеющие слотов с присоединенными процедурами F_p (они представляют фреймы-структуры, аналогичные таблицам, из которых только можно брать информацию в процессе вывода, но изменение значений слотов на время вывода запрещено).

К активным фреймам относятся системные фреймы, например, фрейм приложения и др., имеющие хотя бы одну присоединенную исполняемую процедуру.

Общая схема решения задачи зависит от того, в какой форме ищется решение:

- а) традиционно, в форме алгоритма по данным пассивных фреймов;
- б) поиском по образцу решения, наиболее близкого к образцу среди пассивных фреймов с использованием той или иной меры близости;
- в) методом распространения возбуждения по сети фреймов, содержащих все возможные решения проблемы в группе конечных фреймов-результатов (экземпляров) по указанным экземплярам входных фреймов-параметров информационного запроса.

В системе должен иметься исходный активный фрейм запроса стандартной структуры, содержащий слоты параметров запроса, а также слот результата.

В системах реального времени (мониторинга, слежения, обработки оперативной информации) фреймы оперативной информации непрерывно пополняются извне или в результате обработки. Однако, любая обработка и получение



промежуточных результатов должна проходить в условиях постоянства заданных параметров в шаговом режиме.

Следует различать базу знаний среды разработки – общую базу знаний, применимую для всех возможных приложений (справочная библиотека) и частную базу знаний (приложение), полностью определяющую состав и структуру частных фреймов-прототипов и экземпляров, специфичных для конкретного приложения.

Фрейм конкретного приложения является в общем случае агрегацией фреймов, входящих в информационную базу (пассивных во время решения задачи и переходящих в активные при вводе (сборе) информации) и активных фреймов логического вывода (в частном случае – одного).

Чаще всего приходится разделять фрейм-ситуацию «запрос» на частные фреймы достижения промежуточных результатов (агрегация фреймов-запросов).

Полная база знаний $K = K_{\text{системы}} + K_{\text{приложений}}$,

$$\hat{E} = \bigcup_{i=1}^N K_{ni},$$

где K_{ni} – частная база знаний i -го приложения.

В состав базы знаний входят знания статические K_s (структуры фреймов, значения слотов на множестве типов \mathbb{T}) и динамические K_d (системные методы и процедуры инструментальной системы, присоединенные процедуры и процедуры-демоны фреймов-приложений)

$$K = \hat{E}_s + K_d = K_{s_sys} + K_{d_sys} + \bigcup_{i=1}^N K_{s_app_i} + \bigcup_{i=1}^N K_{d_app_i},$$

где индекс sys относится к инструментальной системе, индекс app_i к i -му приложению.

Фрейм как единица представления знаний отображает как статические, так и динамические знания, причем все множество фреймов F , эквивалентное знаниям K , представляется в виде :

$$F = F_{sys} + F_{app1} + F_{app2} + \dots + F_{appN},$$

где F_{app_i} – множество фреймов i -го приложения (множества могут пересекаться).

Рассмотрим простой случай одного приложения, тогда $F = F_{sys} + F_{app}$

Концептуально все множество фреймов представляет собой многоуровневую иерархию обобщений, где верхний уровень является единственным фрейм-прототип $f_0^0 \in F_{sys} \subseteq F$, а терминальные (фреймы нижнего уровня) $f_1^k, f_2^k, \dots, f_i^k, \dots, f_N^k$ – фреймы-экземпляры (k – № нижнего уровня) являются наследниками (потомками) фреймов предыдущих уровней, наследуя от них статические и динамические знания, $f_i^k \in F$.

Фрейм рассматривается как некоторый контейнер (пакет) знаний, имеющий имя (идентификатор) $id \in I, I \subset \mathbb{I}$, (\mathbb{I} – полное множество идентификаторов фреймов и слотов) и содержащий как агрегацию набор слотов. Каждый слот s



является также носителем статических и динамических знаний в виде возможных значений, определенных на системе типов \mathbb{T} , и набора присоединённых процедур из некоторого множества выражений \mathbb{E} , λ -выражений $\Lambda(\mathbb{T})$; $\mathbb{E} \subseteq \Lambda(\mathbb{T})$ [6]. В частном случае λ -выражение может включать константы, в том числе имена присоединенных процедур, а также ссылки на другие слоты и значения NIL (неизвестно).

Для формализации фреймовых структур будем (аналогично [6]) рассматривать функцию состояний:

$W: I \rightarrow F$, отображающую множество идентификаторов в множество фреймов (эта функция фиксирует состав всей базы знаний на текущий момент); $W \in \mathbb{W}$, где \mathbb{W} - множество возможных состояний.

Обозначим через $W_{IA} \in \mathbb{W}$ состояние информационной базы приложения, которое определяет часть фреймов (системных и приложения), пассивных по отношению к логическому выводу, а через $W_{CA} \in \mathbb{W}$ – состояние системы управления выводом (решателя) – набора активных фреймов.

Каждый фрейм $f \in F$ является функцией $f: I_f \rightarrow S$, отображающей множество идентификаторов слотов фрейма в множество слотов S . Динамические знания представляются множествами демонов и присоединенных процедур и правил. Состояние W_{IA} информационной базы приложения на период вывода считается неизменным.

Считаем, что фреймы включают слоты, задаваемые в общем случае кортежами:

$$S = \langle v, u, \mathcal{A}_i, \mathcal{A}_j, \mathcal{A}_k, \alpha \rangle,$$

$v \in \mathbb{T}$ – значение слота; $u \in \mathbb{T}$ – значение слота по умолчанию; \mathcal{A}_i – упорядоченное множество присоединенных к слотам процедур-демонов поиска значений слота типа IF_NEEDED; \mathcal{A}_j – упорядоченное множество присоединенных к слоту процедур-демонов типа IF_CHANGED, обрабатывающих событие изменения значения слота; \mathcal{A}_k – упорядоченное множество ограничений на значения слота (набор правил или предикатов $C_k \in \mathbb{E}$); α – флаг, используется в контексте вывода для управления выводом, булевого типа.

Для доступа к значениям слота используем операцию разыменования:

$$s.v, s.u; s.IF_NEEDED_i; s.\alpha \text{ и т.п.}$$

Считая идентификаторы фреймов уникальными во всей системе, а идентификаторы слотов – уникальными в пределах фрейма, в операции разыменования объекты f и s заменяем их соответствующими идентификаторами и в этом контексте мы можем считать, что запись $f.s$ одновременно является идентификатором (адресом) слота s во фрейме f .

В первоначальном состоянии (на нулевом шаге вывода) $f.s.v = \text{NIL}$ для всех слотов всех фреймов подсистемы вывода, а также $f.s.\alpha = \text{NIL}$.

Операции присваивания значения слоту будем рассматривать как функцию, в общем случае осуществляющую изменение состояния:



- а) системы в целом;
- б) информационной базы в режиме только пополнения длительно хранимых знаний;
- в) только подсистемы вывода в процессе получения результата решения задачи при неизменном состоянии информационной базы.

Это соответствует изменению состояний W_{IA} и W_{CA}

- а) $\langle := \rangle: I \times T \rightarrow T \rightarrow W$;
- б) $\langle := \rangle: I_{IA} \times T \rightarrow W_{IA} \rightarrow W_{IA}$ (приобретение знаний);
- в) $\langle := \rangle: I_{CA} \times T \rightarrow W_{CA} \rightarrow W_{CA}$ при $W_{IA} = \text{const}$ (решение задачи).

Здесь $W_{IA} \subseteq W$, $W_{CA} \subset W$, $W_{IA} = W$ при отсутствии подсистемы вывода.

Важнейшими иерархическими фреймовыми структурами являются обобщения с наследованием и агрегации. Для их образования в определение фрейма вводятся специальные системные слоты со значениями ссылочного типа.

- 1) $f g \Leftrightarrow \| f.isa \| = g$, где $\| \cdot \|$ - операция вычисления значения системного слота потомка $f.isa$ (обобщение, g - родитель) .
- 2) $f h \bowtie \| f.part_of \| = h$ (агрегация, f входит в h как составная часть).

Изложенный подход и нотация могут быть использованы в процессе разработки языка представления знаний и обоснования стратегий решения задач в интеллектуальной информационной среде с унифицированной фреймовой базой знаний.

Литература

1. Дерябкин В.П. База знаний системы синтеза и параметрической настройки проблемно-ориентированной информационной компьютерной среды // Перспективные информационные технологии в научных исследованиях, проектировании и обучении «ПИТ-2006», Т.1. – Самара, СГАУ, 2006. – с. 65 – 69
2. Дерябкин В.П. Среда визуальной разработки интеллектуальных систем продукционно-фреймового типа // Математическое моделирование информационных процессов и систем в науке, технике и образовании. – Самара, СГАСУ, 2010. – с. 48 – 51
3. Дерябкин В.П., Белоусов А.И. Фреймовая база знаний информационной компьютерной среды // Перспективные информационные технологии для авиации и космоса (ПИТ-2010). Избранные труды Международной конференции с элементами научной школы для молодёжи. – Самара, СГАУ, 2010. – с. 61 – 64
4. Minsky M. A Framework for representing knowledge. – Cambridge: MIT Press, 1974
5. Джарратано Д., Райли Г. Экспертные системы: принципы разработки и программирования, 4-е издание. – М.: ООО «И.Д. Вильямс», 2007. – 1152 с.
6. Сошников Д.В. Логический вывод на основе удалённого вызова и включения в системах с распределённой фреймовой иерархией / под ред. В.Е. Зайцева. – М.: «Вузовская книга», 2002. – 48 с.